ITS Technical Bulletin 220
IBM TCP/IP FTP

Issued Date: 25 Aug 1995
Effective Date: 25 Aug 1995
Section/Groups: Security/Software Support
Submitted By: Farrell Wiser
Approved By: Clair Christensen

FTP is usually run interactively by starting and entering commands from your terminal. You can also run FTP as a batch job but you must supply the job control language (JCL) file. You can use batch when you know what functions you want to perform, when you want a hard copy of the results, or when you want to perform an FTP function many times.

The file containing the JCL cannot have sequence numbers in it. Input streams containing sequence numbers cause unpredictable results because the FTP client parses the job stream at the end of record (EOR).

Caution: Do not attempt to use server parameters and options on this JCL, because it is acting as a client.

Warning: Because MVS batch cannot display a return code greater than 4096, the return code displayed by the JCL will probably not match the exit return code that FTP displays in its message. What is displayed is the result of a modulo divide of the FTP exit return code by 4096. That is, the return code displayed is the remainder of the FTP exit return code divided by 4096.

See the examples on the following pages.
Following is an example of a 'STEP' in a 'BATCH' JOB that performs an FTP.

```
  //anyname      JOB parms,acct,msgclass,PARM='.....',CLASS=,etc...
 .....                                                  /=== Or bigger, for LARGE filesizes....
  //STEPxx      EXEC PGM=FTP,REGION=2048K
  //SYSPRINT  DD SYSOUT=*
  //OUTPUT     DD SYSOUT=*
  //NETRC    DD  DSN=(useridhlq).NETRC,DISP=SHR       <==== Not normally used here...
  ///INPUT    DD DISP=SHR,DSN='your dsname'.....which would contain:
                ( FTP commands of your choice):    type e
                                                       mode b
                                                       put idss.parts
                                                       get anyname
                                                       /*
  //*
  //*       TRANSFER THE  FILE 'DOWN' TO THE XX machine   ( a 'PUT')
  //*
         - OR-
```

```
  //INPUT    DD  *
    type e
    mode b
    put idss.parts
         - OR -
  //INPUT    DD DISP=SHR,DSN='your dsname' (which would contain:
                          (FTP commands, such as):        type e
                                                          mode b
                                                          get idss.parts
                                                          /*              )
         - AND/OR -
    //anyname   DD DISP=(NEW,CATLG),DSN='your dsname', <=== this DD will 'alllocate'
    //              UNIT=SYSDA,SPACE=(xxx,(xx,x),RLSE),              a file to 'receive' the data
    //          DCB=(LRECL=xxx,BLKSIZE=xxxx,RECFM=xx)      'FROM' the XX machine..(a
'GET')
```

In the previous example, REGION=2048K is a minimum requirement. The requirement could increase depending on the block size of the data set being transmitted.

The first JCL statement is a standard job statement. You should use a 'CLASS' for execution that will run it on the CPUn you need, as this FTP process can be run on either CPU2 or CPU4.

The next JCL statement is an EXEC statement that specifies PGM=FTP and a region parameter since FTP might use more storage than your default region size.

Note: For PARM=, you can specify any parameter that is valid when invoking FTP from your terminal. See "FTP Command--Enter the FTP Environment" in related manuals, for more information.

To run FTP in batch, you MUST have at least the following three DD statements:

//SYSPRINT DD Specifies the data set where you want messages to be returned.

//INPUT DD Specifies the data set where the FTP subcommands to be performed are located.

//OUTPUT DD Specifies the data set where FTP is to place the output messages of the FTP functions performed.

Note: The data set specified on the OUTPUT DD statement should have an LRECL of 160 with any block size that is a multiple of the LRECL. The data set specified on the INPUT DD statement should have an LRECL of 80 with any block size that is a multiple of the LRECL.

You can use the user_id.NETRC data set, as defined by the //NETRC DD statement in Figure 1,

to identify the userid and password for a batch-processed remote login. You can also specify the userid and
password in the //INPUT DD statement.

Figure 2 shows the records in an INPUT DD data set that contains the FTP commands to be executed.

```
*
* hostname
* userid  passwd
* dir
* put myfile.listing
* quit
*
```

The first line of Figure 2 contains the name of the host that you want FTP to use. The second line contains the user ID to which you want to connect, followed by its password. The next 3 lines contain the FTP commands that you want FTP to perform. In this example, FTP is doing a directory listing of the server to which you are connecting. The example then instructs FTP to send a file to the server. The last line ends the connection.

The results of the session and any commands you execute appear in the //OUTPUT DD data set. The //SYSPRINT DD can contain some additional messages that relate to the execution of your FTP session.

Note: If you do not want your password to be copied to the output file, specify your user ID and password on separate input lines. See the example in the related manual.

Addtional information can be found in: TCP/IP User's Guide, Version 3 Release 1, IBM Form # SC31- 7136-0; online BookManager Bookname = EZAA2300, Chapter 3: Using FILE TRANSFER, 'Submitting FTP Requests in Batch'